

**IN THE CLAIMS:**

1. (Currently Amended) A method of managing memory in a computing device, comprising:
  - receiving a notification of a runtime correctable error associated with a memory cell,  
wherein the receiving a notification step further comprises:
    - determining if any memory extents in the memory have not been memory scrubbed;
    - if memory extents exist that have not been memory scrubbed, identifying a next memory extent that has not been memory scrubbed;
    - scrubbing the next memory extent to identify any memory cell errors;
    - determining if any redundant bit lines are available for redundant bit line steering;
    - and
    - receiving the notification of the runtime correctable error if redundant bit lines are available for redundant bit line steering;
  - determining if the runtime correctable error has persisted for longer than one memory scrub cycle; and
  - requesting dynamic memory page deallocation for a page of memory associated with the memory cell with which the runtime correctable error is associated if the runtime correctable error has persisted for longer than one memory scrub cycle.
2. (Original) The method of claim 1, wherein determining if the runtime correctable error has persisted for longer than one memory scrub cycle includes:
  - comparing an address of the memory cell with addresses in a history correctable error table; and
  - determining that the runtime error has persisted for longer than one memory scrub cycle if the address of the memory cell is found in the history correctable error table.
3. (Original) The method of claim 2, wherein the history correctable error table is updated after each memory scrubbing operation cycle is completed.

4. (Currently Amended) The method of claim 2, further comprising:  
determining if the address of the memory cell is present in a current [[CE]] correctable error table identifying memory cells that ~~are~~ have a correctable error identified in a current memory scrub cycle; and  
incrementing a count for a matching entry in the current [[CE]] correctable error table if the address of the memory cell is present in the current [[CE]] correctable error table.
5. (Currently Amended) The method of claim 4, further comprising:  
if the address of the memory cell is not present in the current [[CE]] correctable error table, determining if the current [[CE]] correctable error table has sufficient capacity for another entry;  
adding an entry to the current [[CE]] correctable error table corresponding to the address of the memory cell, if the current [[CE]] correctable error table has sufficient capacity; and  
setting a count associated with the entry to 1.
6. (Original) The method of claim 1, further comprising:  
comparing a total number of correctable errors to a predetermined threshold; and  
masking off further correctable error notifications until a current memory scrub cycle completes if the total number of correctable errors is equal to or greater than the predetermined threshold.
7. (Canceled)
8. (Currently Amended) A computer program product in a computer readable medium for managing memory in a computing device, comprising:  
first instructions for receiving a notification of a runtime correctable error associated with a memory cell, wherein the first instructions further comprise:  
first sub-instructions for determining if any memory extents in the memory have not been memory scrubbed;  
second sub-instructions for identifying a next memory extent that has not been memory scrubbed, if memory extents exist that have not been memory scrubbed.;

third sub-instructions for scrubbing the next memory extent to identify any memory cell errors;

fourth sub-instructions for determining if any redundant bit lines are available for redundant bit line steering; and

fifth sub-instructions for receiving the notification of the runtime correctable error if redundant bit lines are available for redundant bit line steering;

second instructions for determining if the runtime correctable error has persisted for longer than one memory scrub cycle; and

third instructions for requesting dynamic memory page deallocation for a page of memory associated with the memory cell with which the runtime correctable error is associated if the runtime correctable error has persisted for longer than one memory scrub cycle.

9. (Original) The computer program product of claim 8, wherein the second instructions for determining if the runtime correctable error has persisted for longer than one memory scrub cycle include:

instructions for comparing an address of the memory cell with addresses in a history correctable error table; and

instructions for determining that the runtime error has persisted for longer than one memory scrub cycle if the address of the memory cell is found in the history correctable error table.

10. (Original) The computer program product of claim 9, wherein the history correctable error table is updated after each memory scrubbing operation cycle is completed.

11. (Currently Amended) The computer program product of claim 9, further comprising:  
instructions for determining if the address of the memory cell is present in a current [[CE]] correctable error table identifying memory cells that ~~are~~ have a correctable error identified in a current memory scrub cycle; and

instructions for incrementing a count for a matching entry in the current [[CE]] correctable error table if the address of the memory cell is present in the current [[CE]] correctable error table.

12. (Currently Amended) The computer program product of claim 11, further comprising:  
instructions for determining if the current [[CE]] correctable error table has sufficient capacity for another entry if the address of the memory cell is not present in the current [[CE]] correctable error table;  
instructions for adding an entry to the current [[CE]] correctable error table corresponding to the address of the memory cell, if the current [[CE]] correctable error table has sufficient capacity; and  
instructions for setting a count associated with the entry to 1.
13. (Original) The computer program product of claim 8, further comprising:  
fourth instructions for comparing a total number of correctable errors to a predetermined threshold; and  
fifth instructions for masking off further correctable error notifications until a current memory scrub cycle completes if the total number of correctable errors is equal to or greater than the predetermined threshold.
14. (Canceled)
15. (Currently Amended) An apparatus for managing memory in a computing device, comprising:  
means for receiving a notification of a runtime correctable error associated with a memory cell;  
means for determining if any memory extents in the memory have not been memory scrubbed;  
means for identifying a next memory extent that has not been memory scrubbed, if memory extents exist that have not been memory scrubbed;  
means for scrubbing the next memory extent to identify any memory cell errors;  
means for determining if any redundant bit lines are available for redundant bit line steering; and  
means for receiving the notification of the runtime correctable error if redundant

means for determining if the runtime correctable error has persisted for longer than one memory scrub cycle; and

means for requesting dynamic memory page deallocation for a page of memory associated with the memory cell with which the runtime correctable error is associated if the runtime correctable error has persisted for longer than one memory scrub cycle.

16. (Original) The apparatus of claim 15, wherein the means for determining if the runtime correctable error has persisted for longer than one memory scrub cycle includes:

means for comparing an address of the memory cell with addresses in a history correctable error table; and

means for determining that the runtime error has persisted for longer than one memory scrub cycle if the address of the memory cell is found in the history correctable error table.

17. (Original) The apparatus of claim 16, wherein the history correctable error table is updated after each memory scrubbing operation cycle is completed.

18. (Currently Amended) The apparatus of claim ~~[[9]]~~ 16, further comprising:

means for determining if the address of the memory cell is present in a current ~~[[CE]]~~ correctable error table identifying memory cells that ~~are~~ have a correctable error identified in a current memory scrub cycle; and

means for incrementing a count for a matching entry in the current ~~[[CE]]~~ correctable error table if the address of the memory cell is present in the current ~~[[CE]]~~ correctable error table.

19. (Currently Amended) The apparatus of claim ~~[[11]]~~ 18, further comprising:

means for determining if the current ~~[[CE]]~~ correctable error table has sufficient capacity for another entry if the address of the memory cell is not present in the current ~~[[CE]]~~ correctable error table;

means for adding an entry to the current [[CE]] correctable error table corresponding to the address of the memory cell, if the current [[CE]] correctable error table has sufficient capacity; and

means for setting a count associated with the entry to 1.

20. (Currently Amended) The apparatus of claim [[8]] 15, further comprising:

means for comparing a total number of correctable errors to a predetermined threshold; and

means for masking off further correctable error notifications until a current memory scrub cycle completes if the total number of correctable errors is equal to or greater than the predetermined threshold.

21. (Canceled)

22. (New) The method of claim 1 further comprising:

determining if a predetermined threshold number of memory cell errors is exceeded during the scrubbing of the memory; and

if the predetermined threshold number of memory cell errors is exceeded, performing redundant bit line steering.

23. (New) The computer program product of claim 8 further comprising:

instructions for determining if a predetermined threshold number of memory cell errors is exceeded during the scrubbing of the memory; and

instructions for performing redundant bit line steering, if the predetermined threshold number of memory cell errors is exceeded.

24. (New) The apparatus of claim 15 further comprising:

means for determining if a predetermined threshold number of memory cell errors is exceeded during the scrubbing of the memory; and

means for performing redundant bit line steering, if the predetermined threshold number of memory cell errors is exceeded.